Disruptive Technologies for Data-Intensive Computing

Nicholas J. Wright, Lavanya Ramakrishnan, R. Shane Canon
Lawrence Berkeley National Lab, Berkeley CA
{njwright, LRamakrishnan, scanon} @lbl.gov

Abstract

Several potentially disruptive technologies are on the horizon which have the potential to transform our abilities to perform data-intensive computations. On the hardware side, flash memory and other upcoming nvRAM technologies have the potential to modify the memory hierarchy, allowing storage of significant quantities of data close to the compute device. On the software side, HDFS, the Hadoop filesystem provides mechanisms to leverage properties such as data locality to process data intensive computations. Together, these disruptive technologies provide a strong foundation for data-intensive computing. In this work-in-progress paper we report upon our ongoing evaluations of these technologies.

Introduction

Across many domains, scientists are struggling with a tsunami of data [3]. Emerging sensor networks, more capable instruments, and ever increasing simulation scales are generating data at a rate that exceeds our ability to effectively manage, curate, analyze, and share it. This has led to the emergence of the field of Data-intensive Computing, and the design of High-Performance Computing systems specifically focused upon meeting the needs of this emerging field [4][5]. Generally speaking, these system designs have focused upon optimizing the I/O capabilities of the machines.

In this work-in-progress paper, we discuss two hardware-based trends that will effect our abilities to create computing systems optimized for data-intensive computing. The first is due to the current multicore era during which the number of cores on a processor is rapidly increasing, leading to reduced memory capacities and bandwidths per core. The second issue is the trends in I/O performance. Increases in storage capacity continue to outpace increases in bandwidth to storage. Consequently, storage systems require thousands of drives to meet the bandwidth required to field balanced systems, which is expensive from both a cost and a power perspective. For data-intensive computing these are worrying trends, as the gap between the ability to generate data and our ability to extract understanding from that data is increasing. At the same time as these trends are occurring there are new non-volatile memory technologies such as NAND flash, Phase Change Memory (PCM), and Magnetic RAM (MRAM) emerging that could mitigate these issues somewhat. Compared to a regular spinning disk, flash has a large latency advantage for both read and write operations making it very attractive for petascale data analytics.

Additionally, software technologies have also recently emerged that address some of the needs of data-intensive science. Hadoop[1], the open source implementation of MapReduce[2], provides a framework for composing and managing highly asynchronous computations. Tools such as Pig, HBase, etc provide a way to store, manage and query data. Hadoop promises to be critical components of a data ecosystem. However there is still a gap in effectively using these new software tools specifically designed for data-

intensive computing especially when one considers them in combination with the disruptive hardware technologies just described.

Thus it is critical to examine and evaluate these new emerging technologies, and how they can be combined effectively. In this paper, we provide early results from our evaluations of these disruptive technologies.

Flash Technology Evaluation

Flash memory, and other emerging nvRAM technologies, such as Phase Change Memory (PCM), have the potential to modify the memory hierarchy of compute devices by providing non-volatile storage with read and write latencies significantly faster than those achievable to storage today. However, there are many unanswered questions about the optimal use of such technologies. For example, to which interface should they be attached, how does the I/O pattern change the performance? To begin to gain an understanding of these issues we evaluated the performance characteristics of several flash memory devices. Our evaluation included five devices, three PCI attached ones and two SATA attached ones. Primarily we were interested in determining the peak bandwidth and IOPS capabilities of the devices. Our results are shown in Table 1.

Table 1 Peak Performance Characteristics of the Flash Devices.

Device	Connection Type	Peak Bandwidth MB/s		I/O (4K) operations per second x10^3	
		Read	Write	Read	Write
Intel X25-M SATA	SATA	200	100	19.1	1.49
OCZ Colossus SATA		200	200	5.21	1.85
FusionIO ioDrive Duo	PCIe-4x	800	690	107	111
Texas Memory Systems RamSan20		700	675	143	156
Virident tachIOn	PCIe-8x	1200	1200	156	118

There are several interesting performance characteristics as compared to a regular spinning disk, and also some interesting differences between the flash devices themselves. For both bandwidth and IO/s, the PCIe devices are clearly much more capable than the SATA attached ones, by a factor of 4-6x in bandwidth and 5-100x in IO/s. Typically a regular SATA hard drive today can support approximately 80 MB/s or 90 IOPs for both read and write. Thus a flash device, especially if it is PCIe attached, can provide a significant IOPS advantage over a traditional hard drive based storage system.

For data-intensive computing there are several implications. Firstly, for IOPs bound operations, such as databases, flash technology can provide a large performance boost, by as much as a factor of a thousand. Secondly, by being used as on-node storage it can provide high-performance I/O capabilities as close as possible to the compute resource. This will enhance the abilities of new computing paradigms for the efficient processing of large volumes of data, such as Hadoop.

Hadoop Technology Evaluation

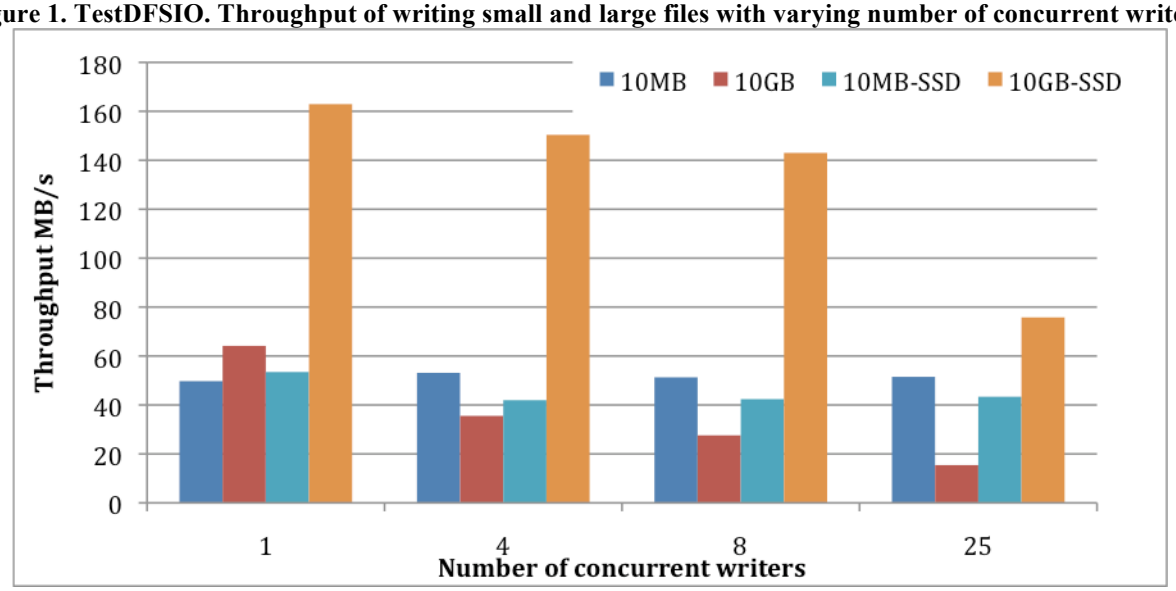
Apache Hadoop is open-source software that provides capabilities to harness commodity clusters for distributed processing of large data sets through the MapReduce model. Inherently Hadoop is designed to be reliable and scalable and uses the Hadoop File System's data location capabilities to manage scheduling. Hadoop supports MapReduce and higher-level tools in Hadoop facilitate customization of the map and reduce functions for specific job types.

The Hadoop File System (HDFS) is the primary storage model used in Hadoop. HDFS is modeled after the Google File system and has several features that are specifically suited to Hadoop/MapReduce. Those features include exposing data locality and data replication. Data locality is a key aspect of how Hadoop achieves good scaling and performance and Hadoop attempts to locate computation close to the data. This is especially true in the Map phase which is often the most I/O intensive phase. Data is transparently replicated for fault tolerance and to provide more opportunities to execute computation near data. The file system continuously monitors the number of replicas and data integrity, and will re-replicate data as needed to ensure quality. The file system also attempts to distribute replicas across failure domains to improve the likelihood of access to data for various failure models. The chief disadvantage of HDFS is that it is primarily accessible through special APIs. These APIs are primarily implemented in Java but bindings for other languages are available. An application must be modified to use these APIs to access data stored in HDFS.

We ran a few benchmark tests to understand the Hadoop File System (HDFS). These tests were run on the 40 node Hadoop installation on the Magellan cloud testbed. Each node consists of two quad-core Intel Nehalem 2.67 GHz processors per node, 8 cores per node, 48 GB DDR3 1066 MHz memory per node and a 1 TB SATA disk or a 250 GB SSD drive (OCZ Colossus) per node. We use the Hadoop TestDFSIO to understand the file system characteristics that measures the I/O performance of HDFS. Figure 1 shows the throughput for a small (10MB) and large file size (10GB) with varying concurrent writers. The default block size is 128MB in our setup. For small file sizes, the throughput remains fairly constant with varying number of concurrent writers. However, the throughput decreases as the number of concurrent writers increases. This is dependant on the HDFS block size and overheads of the file system. HDFS atop SSD drives gives better throughput for large files.

Hadoop promises to be a disruptive technology for data intensive applications, however, there are challenges in hardware and software that need to be addressed. Data-intensive scientific applications vary in their I/O usage patterns, from a large number of small files to a small number of very large files. This will require careful consideration in managing data across different hardware technologies in the context of the application.

Figure 1. TestDFSIO. Throughput of writing small and large files with varying number of concurrent writers



Summary

In this paper we describe our preliminary results from evaluating these two new technologies. Flash memory and technologies such Hadoop promise to be enabling technologies for building next-generation data-intensive computing systems. In future work we will look at using them with scientific applications to observe the effect they have in those cases upon performance.

Acknowledgements

This work was funded in part by the DOE award DE-FC02-06ER25767 the Petascale Data Storage Institute and by the Advanced Scientific Computing Research (ASCR) in the DOE Office of Science under contract number DE-AC02-05CH11231. This research used resources of the National Energy Research Scientific Computing Center, under Contract No. DE-AC02-05CH11231.

References

1. Apache Hadoop http://hadoop.apache.org/.
2. J. Dean, And S. Ghemawat. MapReduce: Simplified Data Processing On Large Clusters. In Osdi'04: Proceedings Of The 6th Conference On Symposium On Opearting Systems Design & Implementation (Berkeley, CA), 2004), Usenix Association, Pp. 10–10.
3. G. Bell et al, "Beyond the Data Deluge," Science, March 2009.
4. J. He, J. Bennett, and A. Snavely, *DASH-IO: an empirical study of flash-based IO for HPC*, IEEE and ACM, Supercomputing 2010, November 13-19, 2010.
5. *Lawrence Livermore Teams with Fusion-io to Redefine Performance Density* http://www.fusionio.com/press/Lawrence-Livermore-Teams-with-Fusion-io-to-Redefine-Performance-Density/